

# Multilevel System as Multigraph<sup>\*</sup>

Waldemar Korczyński<sup>1</sup>, José de Jesús Cruz Guzmán<sup>2</sup>, and  
Zbigniew Oziewicz<sup>2\*\*</sup>

<sup>1</sup> University of Arts and Science, ulica Wesola 52 PL - 25353 Kielce, Poland  
korwald@wsu.kielce.pl

<sup>2</sup> Universidad Nacional Autónoma de México,  
Facultad de Estudios Superiores Cuautitlán, Apartado Postal 25, 54700 Cuautitlán  
Izcalli, Estado de México {cruz,oziewicz}@servidor.unam.mx  
oziewicz@ift.wni.wroc.pl

**Abstract.** Graph based models of hierarchical systems are usually seen as "graphs equipped with some refinements", understood as the homomorphisms or (bi)simulations. In such a model it is not possible to consider phenomena happened on different levels of the system. We propose a new formalism of multi-graphs allowing to see a hierarchical system similar as a formula of second order logic, i.e. to consider all levels "at the same time". The concurrency in hierarchical system is modelled in terms of multi-graphs.

## 1 Introduction

Hierarchical systems are usually described by refinement of some of its elements namely modules.(W. Korczyński 2000, [6]) This way of thinking about hierarchical systems as a sum of its modules leads to consideration of any part of the system on another abstraction level which make impossible to consider at the same time properties of parts of systems being of different levels of abstraction (that could be comparable by means of the hierarchical order).

In the paper we propose a way of seeing hierarchical systems, which allow to consider the elements of all levels of abstractions in exactly the same way. The ideology can be seen as a generalization of the notions of the graph ( $\equiv$  one-graph). Instead of considering two levels of abstraction, namely vertices and edges, we can treat a hierarchical system as a sequence of elements, called in these context cells, connected by operations having exactly the same properties as the well known operations of source and target in a one-graph. The structure thus obtained is called multi-graph or more precisely an n-graph. The aim of the paper is to describe the construction of n-graph based on some typical presentations of graphs.

---

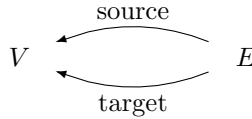
<sup>\*</sup> Supported by el Consejo Nacional de Ciencia y Tecnología (CONACyT) de México, proyecto # U41214-F, and by UNAM, DGAPA, Programa de Apoyo a Proyectos de Investigacion e Innovacion Tecnologica, proyecto # IN 105402.

<sup>\*\*</sup> Zbigniew Oziewicz is a member of Sistema Nacional de Investigadores, México, No de expediente 15337.

The paper is organized as follows: Section 1 are presented the principal aims of these work. Section 2 recall some well known definitions of graphs. Section 3 are proposed some interpretations on n-graphs as models of hierarchical systems. Section 4 a extension of n-graphs into n-category as been described. Section 5 category structure is presented for refining concurrency of processes in multi level systems. Section 6 some historical remarks are presented. The paper is written in the language of category theory. We do not assume the reader is familiar with the category theory because the paper is in essence self-container; we explain all the notions at the very beginning level.

## 2 Some Presentations of Graphs

A directed graph can be seen as a sequence  $s, t : V \longrightarrow E$  being the source and the target,



**Fig. 1.** A directed graph.

The elements of the set  $V$  are called vertices of the graph and are illustrated as the zero-dimensional points. The elements of  $E$  are called arrows or edges of the graph and are illustrated as one-dimensional directed segments of some lines. However more adequate terminology is 0-cell  $\in V$  and 1-cell  $\in E$  (A. Burroni 1981, [3]). A graph (Figure 1) is said to be a 1-graph. In the theory of automata there is also alternative terminology: a state or 0-transition  $\in V$  and 1-transition  $\in E$ . (A. Obtułowicz 2001, [12])

In some situations it is suitable to consider a graph consisting of ‘arrows’ only (a ‘globular’ graph).

**Definition 1 (One-sorted graph).** Let  $id, s, t : X \longrightarrow X$  be an alphabet of a monoid,  $w$  be the homogeneous word in this alphabet and  $|w| \in \mathbb{N}$  be a length of  $w$ . A graph (a 1-graph) is defined by (globular) relations

$$\forall |w| = 1, \quad s \cdot w = w, \quad \& \quad t \cdot w = w.$$

This presentation of a graph was originated from Hasse & Michler (M. Hasse & Michler L. 1966, [5]). We call it one sorted presentation of a graph, or a globular graph.

Let  $X = E \cup V$ . One can extend domain of  $s$  and  $t$  on Figure 1, from  $E$  to  $X$  by  $s|V = id$  and  $t|V = id$ . Then  $s, t : X \longrightarrow X$  is a one sorted graph.

**Definition 2 (Reflexive graph).** A graph  $(V, E, s, t)$  is said to be reflexive if there is an injective section  $i : V \longrightarrow E$ ,  $s \circ i = t \circ i = id_V$ . A reflexive graph is an algebra  $(V, E, s, t, i)$ .

Another presentation of graphs defines them as (indexed) families of sets.

**Definition 3 (Graph as a family of sets).** *A graph is a map  $G : V \times V \longrightarrow 2^E$ . The elements of  $V$  are called the vertices and those of the disjoint union*

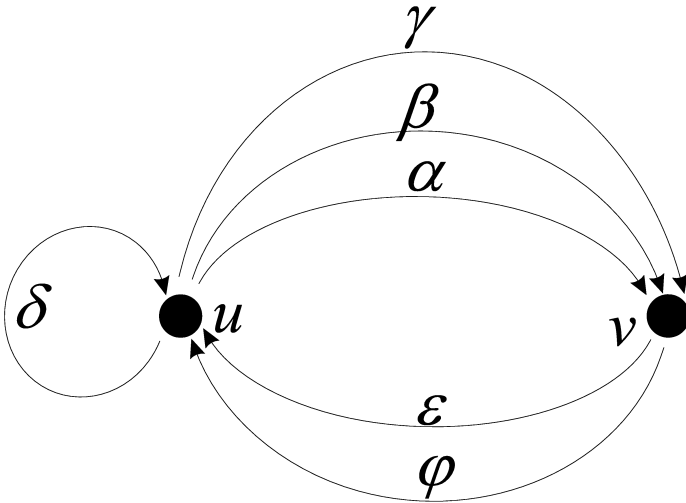
$$\text{arrows}(G) = \bigsqcup_{(u,v) \in V \times V} G(u,v)$$

*the arrows of  $G$ . For any  $u, v \in V$  the elements of the set  $G(u,v)$  are called arrows from  $u$  to  $v$  and for any  $\alpha \in G(u,v)$  the vertex  $u$  is called the source and  $v$  the target of  $\alpha$ .*

*The sets of the family  $(G(x,y))_{x,y \in V}$  are pointwise disjoint and we use the standard set-theoretical union instead of the direct sum of sets.*

*Example 1.* Consider the following graph with two vertices: In the following picture (figure 2) the sets  $G(x,y)$  are illustrated. We have here  $G(u,u) = \{\delta\}$ ,  $G(u,v) = \{\alpha, \beta, \gamma\}$ ,  $G(v,u) = \{\varepsilon, \varphi\}$ ,  $G(v,v) = \emptyset$ . The set of vertices of this graph is  $V = \{u, v\}$  and the set of arrows

$$\begin{aligned} \bigsqcup_{(x,y) \in V \times V} G(x,y) &= G(u,u) + G(u,v) + G(v,u) + G(v,v) = \\ &= \{\delta\} + \{\alpha, \beta, \gamma\} + \{\varepsilon, \varphi\} + \emptyset = \{\alpha, \beta, \gamma, \delta, \varepsilon, \varphi\}. \end{aligned}$$



**Fig. 2.** A graph pictured "traditionally"

The above definitions allow to see a graph as a presheave. The corresponding categories are closed under the formations of product, homomorphic image and taking subalgebras.

Graphs are tools for describing various phenomena. Let us list some of them:

- the arrows of a graph are processes and vertices are situations being their initializations (births) and deads.
- schemes of connections, schemes of electrical or telephonic networks, flow-charts (flow diagrams in computer sciences) or (dependence) relations in sets of institutions or persons.
- schemes of dynamical systems. The vertices  $G_0$  of a graph  $G \equiv G_0 \Leftarrow G_1$  are resources, and the arrows  $G_1$  are interpreted as activities (events) consuming these resources.

### 3 Multi-graphs

Graphs are a good description tool for dynamical systems with processes running on one level only. The systems, one meets in many branches of every day life, do not have this property. Typical examples are systems of management where one has to consider the control on the fundamental (lowest) level, the control of processes running on this fundamental level, the control of control processes and so on.

*Example 2 (Medical).* An example in medicine: process running in a biological cell, processes running in an organ, in a group of organs and at the level of the whole organism. A biological cell  $c \in G_1$  in an organ  $O$ : the situation  $sc \in G_0$  is its birth and  $tc \in G_0$  its death. The process  $\alpha \in G_2$  in the organ  $O$  is an arrow of a graph having situations being cells,

$$G_1 \ni c_1 \xrightarrow{\alpha} c_2 \in G_1.$$

In a new 1-graph  $G_1 \Leftarrow G_2$  vertices  $G_1$  are arrows of a 1-graph  $G_0 \Leftarrow G_1$  illustrating processes in the biological cells,

$$\text{vertices}(G_0) = V_0 \xleftarrow{d_0^0} A_0 = \text{arrows}(G_0) = \text{vertices}(G_1) \xleftarrow{d_1^1} A_2 = \text{arrows}(G_1)$$

Continuing this reasoning we obtain a sequence of directed 1-graphs  $G_i \Leftarrow G_{i+1}$ ,  $0 \leq i \leq n-1$ , called a directed n-graph, describing different levels of an organism.

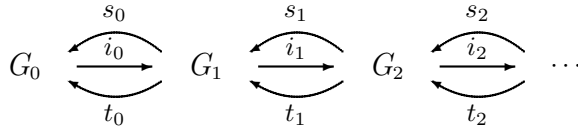
We formalize the above considerations as follows.

**Definition 4 (n-graph).** A directed n-graph  $G$  is a sequence of families of  $i$ -cells  $\{G_i\}$ ,  $0 \leq i \leq n$ , such that  $G_i \Leftarrow G_{i+1}$  is a directed 1-graph with a source map  $s_i : G_{i+1} \rightarrow G_i$  and target map  $t_i : G_{i+1} \rightarrow G_i$ , (Z. Oziewicz 2003, W. Lawvere 1989, [13], [8])

$$V_0 \xleftarrow{d_0^0} A_0 = V_1 \xleftarrow{d_1^1} A_1 = V_2 \xleftarrow{d_2^2} A_2, \dots, V_n \xleftarrow{d_n^n} A_n.$$

The corresponding definition in the case of reflexive graphs can be the following one.

**Definition 5 (n-graph from reflexive graphs).** A directed  $n$ -graph is said to be reflexive if there are sections  $\{i_i : G_i \longrightarrow G_{i+1}\}$ , satisfying the conditions  $s_i \circ i_i = id_{G_i} = t_i \circ i_i$ .



Homomorphisms of  $n$ -graphs are defined componentwise.

**Definition 6.** A homomorphism  $f \in \text{hom}(G, H)$  of an  $n$ -graph  $G$  into an  $n$ -graph  $H$  is a sequence  $f \equiv \{f_i : G_i \longrightarrow H_i\}$ ,

$$\begin{array}{ccc} G_i & \xleftarrow{\quad} & G_{i+1} \\ f_i \downarrow & & f_{i+1} \downarrow \\ H_i & \xleftarrow{\quad} & H_{i+1} \end{array}$$

such that for any  $i \in \mathbb{N}$ ,  $(f_i, f_{i+1})$  is a homomorphism of 1-graphs.

Let us note a difference between the approach used in the above example and the standard modular modelling of multilevel systems. Seeing a system as an  $n$ -graph we can consider it on all levels *at the same time* (A. Obtułowicz 2001, [11], [12]). This point of view is similar to that known from the high order logic when one has to consider variables being individuals, sets of individuals, sets of sets of individuals, etc. Seeing a system as an object consisting of some other objects called in this context modulus we are always at the same level. Refining of a model doesn't lead to a new situation; we are still at the same level because one hasn't to consider any relation between objects of different levels. The last is characteristic for  $n$ -graphs and higher order logic. Let us consider an analogous construction for graphs seeing as families of sets. Having a graph one can consider the family  $\Gamma(G)$ , of all graphs with vertices being arrows in  $G$ , i.e. graphs of the form

$$(G(\alpha, \beta))_{(\alpha, \beta) \in \text{arrows}(G) \times \text{arrows}(G)}$$

This construction can be repeated several times and one can for a given graph  $G_0$  obtain a sequence of families of graphs

$$G_0, \Gamma(G_0), \Gamma^2(G_0) = \Gamma(\Gamma(G_0)), \dots, \Gamma^n(G_0)$$

where  $\Gamma(\Gamma(G))$  is the family of all graphs obtained by the operation  $\Gamma$  from the elements of the family  $\Gamma(G_0)$ ,  $\Gamma(\Gamma(\Gamma(G_0)))$  denotes the family of graphs obtained in the same way from the elements of the family  $\Gamma(\Gamma(G_0))$  and so on.

Fact 1. For any sequence

$$G = G_0, G_1, G_2, \dots, G_n$$

of graphs such that for  $i < n$  it holds  $G_i \in \Gamma^i(G_0)$  we have  $\text{arrows}(G_i) = \text{vertices}(G_{i+1})$ .

**Definition 7.** By an  $n$ -graph we mean any sequence

$$G_0, G_1, G_2, \dots, G_n$$

of graphs satisfying the condition  $G_{i+1} \in \Gamma^i(G_i)$  for any  $i$ . The elements of the sets

$$V_0, V_1, V_2, \dots, V_n, E_n (= V_{n+1})$$

will be called cells, more precisely 0-cells (those from  $V_0$ ), 1-cells (those belonging to  $V_2$ ), ...,  $n$ -cells (the elements of  $V_{n+1}$ ).

*Remark 1 (Normal graph).* For a reflexive graph  $G$  we have

$$\text{vertices}(G) \subseteq \text{arrows}(G)$$

In other words in the above definition we make no constraints on the function  $G : V \times V \rightarrow \text{Set}$ . Particularly we do not exclude the case when for some arrow  $\alpha$  and vertex  $u$  of a graph  $G \in \Gamma^i(G_0)$  the set  $G(\alpha, u)$  is not empty.  $n$ -graphs in which for any arrow  $\alpha$  and vertex  $u$  it holds  $G(\alpha, u) = \emptyset$  for any  $i \leq n$  and  $G \in \Gamma^i(G_0)$  will be called normal.

*Example 3.* In the following picture a 2-graph is illustrated. In this case  $G_0$  is the graph pictured in Figure 3. The operation  $\Gamma$  adds 4 sets of arrows:  $G(u, \delta)$ ,  $G(u, \alpha)$ ,  $G(\alpha, \varepsilon)$ ,  $G(\varphi, \gamma)$ . In the picture only nonempty sets of arrows has been illustrated. So the considered graph is not normal.

## 4 Multi-category

An  $n$ -graph is a carrier for an  $n$ -category by adding of compositions (W. Marcinek & Oziewicz Z. 2001, [10]) as a partial functions  $\circ_i$ , of the form

$$\circ_i : \text{arrows}(G_i) \times \text{arrows}(G_i) \rightarrow \text{arrows}(G_i)$$

being the set theoretical union of total functions of the form

$$G_i(u, v) \times G_i(v, w) \rightarrow G_i(u, w)$$

for  $u, v, w \in \text{vertices}(G_i)$ . (that means  $u, v, w \in G_{i-1}$ ). Such a composition is assumed to be associative and having two sided identities  $id_u$  and  $id_v$  such that for any arrow  $\alpha \in G_i(u, v)$  it holds

$$id_u \circ_i \alpha = \alpha \circ_i id_v = \alpha$$

In other words we make any of graphs

$$G_0, G_1, G_2, \dots, G_n$$

a multiplicative graph with associativity and having two sided identities multiplication, i.e. a category. For any  $i \leq n$  the category thus obtained will be denoted by  $C(G_i)$ .

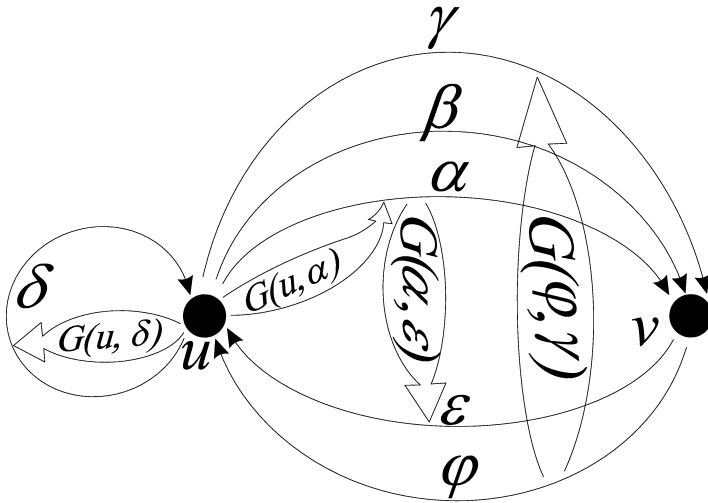


Fig. 3. A 2-graph on the graph picture

**Definition 8** (**n-category** (W. Marcinek & Oziewicz Z. 2001, [10])). An *n-graph*  $G$

$$V_0 \rightrightarrows E_0 = V_1 \rightrightarrows E_1 = V_2 \rightrightarrows E_2 = V \rightrightarrows \dots, V_n \rightrightarrows E_n$$

is said to be an *n-category* we mean any sequence

$$C(V_0 \rightrightarrows E_0), C(V_1 \rightrightarrows E_1), \dots, C(V_n \rightrightarrows E_n).$$

n-categories and some of them applications have been described in Hasse & Michler (M. Hasse & Michler L. 1966, [5])

## 5 Refining and Concurrency

Refining is a method for system describing and analysis. An advantage of this way of treating complicated composed systems is that we are always on the same level of description. We use in any moment the first order logic. Sometimes this approach to systems is not suitable. Typical example of such situations can be found while considering concurrency of processes running in different levels of a hierarchical system. This kind of concurrency can not be described by the modular hierarchical modelling point. One of the examples of such systems can be the biological system mentioned early. Other example can be found in hierarchical management system. Let us consider such a system consisting of two levels. The first one, let us call it production, can be seen as a really production system in a factory, in an office or in economical organization. The second level manage the processes in the first one. So the way of working of such a system

can be seen as follows. Firstly, an object, let us call it  $m$ , performs a planning action  $A$ . As a result one obtains a project, say  $p$ , which will be realized by an object  $w$  on the level 1. Now, one can consider a concurrent activities  $A'$  and  $p$ . Next another project, say  $p'$  being the result of the activity  $A'$ , can be realized concurrently with another designing activity  $A''$ , and so on. The generalization of this concurrency to multilevel systems is evident.

This kind of concurrency is different from the concurrency from Petri nets where one considers concurrency in the set of elements of the same level (B. Baumgarten 1990, M. Leszak and H. Eggert 1988, M. van Sinderen, L. Ferreira, C. A. Vissers and J. Katoen 1995, [1], [9], [14]). The category structure (given by the multiplication of arrows) allows for considering a hierarchy of sequential systems. In this case the only concurrency is the concurrency described above. But one can also consider another kind of concurrency - the concurrency of the same level of the system. This concurrency can be modelled by introducing in the categories corresponding to the levels of the system a new operations expressing the parallel composition of the elements (arrows and vertices) of the system. It can be done in the same way as it has been done in Korczyński 2001 (W. Korczyński 2001, [7]). This leads to the notion of (partial) monoidal n-category. We do not consider here these methods.

## 6 Historical Remark

Some notions used in the paper are not very popular and frequently used. Let us recall the origin of them. The one sorted presentation of graphs has been introduced by Maria Hasse & L. Michler in 1966 (M. Haese & Michler L. 1966, [5]). One can say it belongs to the language of the so called "French school of category theory" originated by Schuetzenberger and Ehresmann (Ehresmann 1950, [4]). Such an approach allows to consider categories as a kind of "partial semigroups" and leads to interesting applications of (semi)group theory seeing as a (meta)language for such branches of mathematics as for example topology. Reflexive graphs have been considered firstly by Burroni (A. Burroni 1981 [3]) Gray and Lawvere (F. W. Lawvere 1989, [8]). The idea was similar to that of introducing one-sorted graphs; one wants to consider categories as a kind of monoids. The embedding of vertices into arrows shows how an "identity" can be introduced into the classical graphs. This identity assigns to every vertex of a graph a "loop" being an arrow having its beginning and end - point in these vertex. Burroni (A. Burroni 1981, [3]) also introduces n-graphs as a tool for describing of some notions in "pure mathematics", more precisely in category theory seeing as a kind of mathematical language. Another interesting papers exploring n - graphs and very interesting applications of n-graphs to some problems in computer sciences have been described by Obtulowicz (A. Obtulowicz 2001, [11], [12]).



## References

1. Baumgarten B., Petri-Netze, Grundlagen und Anwerdungen, Wissenschaftesverlag, Mannheim, 1990
2. Brown R. and P. J. Higgins, On the algebra of cubes, *Journal of Pure and Applied Algebra* **21** (3) (1981) 233–260
3. Burroni Albert, Algèbres graphiques, *Cahier* (1981)
4. Ehresmann, Les Connections infinitesimales dans un espace fibre differentiable, *Coloque de Topologie, Bruxelles* (1950).
5. Hasse Maria and L. Michler, *Theorie der Kategorien*, VEB Deutscher Verlag der Wissenschaften, Berlin 1966
6. Korczyński Waldemar , *Motody Sieciowe - Elementy Filozofii Podejścia*, Kielce 2000
7. Korczyński Waldemar, On some presentations of graphs and hypergraphs, *Instytut Matematyki PAN*, preprint 2001
8. F. William Lawvere, Display of Graphics and their applications, as exemplified by 2-categories and the Hegelian "taco", State University New York at Buffalo, typescript, April 1989.
9. Leszak M., Eggert H., *Petri Netz Methoden und Werkezeuge*, Springer Informatik Fachberichte, 197, Berlin, 1988
10. Marcinek Władysław and Zbigniew Oziewicz, *Miscellanea Algebraicae* (Kielce), Rok 5, No 2/2001, ISBN 83-87798-23-1; math.CT/0104136
11. Obtułowicz Adam, On n-dimensional graphs and n-graphs, *Miscellanea Algebraicae* (Kielce), Rok 5, No 1/2001, pp. 87–96, ISBN 83-87798-22-3
12. Obtułowicz Adam, A note on the connections of automata and weak double categories, *Miscellanea Algebraicae* (Kielce), Rok 5, No 1/2001, pp. 75–86, ISBN 83-87798-22-3
13. Oziewicz Zbigniew, Operad of graphs, convolution and quasi-Hopf algebra, *Contemporary Mathematics* 2003, in press
14. Marten van Sinderen, Luís Ferreira Pires, Chris A. Vissers, Joost-Pieter Katoen, A design model for open distributed processing systems, *Computer Networks and ISDN Systems*, **27** (1995) 1263–1285